ICTNET at Entity Track TREC 2010

Lei Cao[12], Lu Bai[12], Xueqi Cheng[1], Jiafeng Guo[1], Hongbo Xu[1], Yue Liu[1], Xiaoming Yu[1]

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190

2. Graduate School of Chinese Academy of Sciences, Beijing, 100190

Abstract

This paper gives an overview of our work for related entity finding which is proposed in TREC 2010 Entity Track. The goal of the Entity Track is to find the entities relevant to a given query from the web corpus. In this paper, we propose a bipartite graph reinforcement model for entity ranking. As is well known, the entities on the web are embedded not only in the natural language text, but also in the tables and lists. Given a query, both the candidate entities and relevant tables/lists are extracted from web documents. Then the candidate entities extracted from unstructured text are ranked based on a probabilistic model. But the result contains a lot of noise. If some candidate entities are in a relevant table/list, they are more relevant to the given query. And Vice versa, if a table/list contains several candidate entities, it is also more relevant to the query. Based on the above intuition, we construct a bipartite graph and then perform a reinforcement algorithm to re-rank the candidate entities.

1. Introduction

The World Wide Web has been growing rapidly as huge knowledge repository which contains so rich information. Traditional information retrieval systems return a list of documents relevant to user's queries. However, users often show more interest in the entities instead of documents given their queries. So it is necessary to study how to automatically find the related entities given a query.

The main task of Entity Track focuses on the related entity finding (REF): given an input entity, by its name and homepage, the type of the target entity, as well as the nature of their relation, described in free text, find related entities that are of target type, standing in the required relation to the input entity [1] .

2. System Architecture

Our approach can be summarized by the following steps:

1. Form the query based on the source entity and narrative
2. Extract the relevant text snippets from the documents and recognize the named entities with the given target type
3. Compute the initial ranking score for each candidate entity based on a probabilistic model
4. Extract the relevant tables/lists which may contain several target entities from documents
5. Construct the bipartite graph based on the candidate entities and relevant tables/lists. And then the reinforcement learning algorithm is performed over the graph to get the final score for each candidate entity.
6. Find the homepage for candidate entities

The following sections will explain some key steps of our approach.

2.1 Query Formation

The query is formed as a keyword set based on the given entity and narrative. Words indicating the relationship should be extracted from the narrative. With a part-of-speech tagger, we parse the narrative to get the verb or noun to form the keyword set. After the initial keyword set is formed, we argument it with synonyms of the keywords from WordNet [2].

2.2 Named Entities Identification

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **NOV 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **ICTNET at Entity Track TREC 2010**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Chinese Academy of Sciences,Institute of Computing Technology,Beijing, 100190,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the Nineteenth Text REtrieval Conference (TREC 2010) held in Gaithersburg, Maryland on 16-19 November 2010. The conference was co-sponsored by the National Institute of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), and the Advanced Research and Development Activity (ARDA).

14. ABSTRACT

This paper gives an overview of our work for related entity finding which is proposed in TREC 2010 Entity Track. The goal of the Entity Track is to find the entities relevant to a given query from the web corpus. In this paper, we propose a bipartite graph reinforcement model for entity ranking. As is well known, the entities on the web are embedded not only in the natural language text, but also in the tables and lists. Given a query, both the candidate entities and relevant tables/lists are extracted from web documents. Then the candidate entities extracted from unstructured text are ranked based on a probabilistic model. But the result contains a lot of noise. If some candidate entities are in a relevant table/list, they are more relevant to the given query. And Vice versa, if a table/list contains several candidate entities, it is also more relevant to the query. Based on the above intuition, we construct a bipartite graph and then perform a reinforcement algorithm to re-rank the candidate entities.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**4**	

After the relevant documents are retrieved, we extracted the relevant text snippets from the documents. Then the Stanford NER Tagger[1] is used to extract the named entities with the given target type from the text snippets. At the same time, we detect the boundary of named entities with the aid of anchor text. After acquiring the initial list of candidate entities, we apply several heuristic rules to remove the duplicated entities. Finally, we get a list of candidate entities with their support snippets.

2.3 Initial Candidate Entities Ranking

After acquiring the list of candidate entities, an initial ranking value is calculated for each of them. We define Q = {e^s, r, T}, where e^s denotes the source entity, r denotes the relationship between the source entity and the target entity, and T denotes the type of target entity. And let e^t denotes the target entity. The candidate entities are ranked by the probability $p(e^t|Q)$. By applying Bayesian Theorem and Chain rules, $p(e^t|Q)$ could be decomposed into the following form:

$$\begin{aligned} p(e^t \mid Q) &\propto p(e^t, Q) \\ &= p(e^t, e^s, r, T) \\ &= p(r \mid e^s, e^t, T)\, p(e^s, e^t, T) \\ &= p(r \mid e^s, e^t, T)\, p(T \mid e^s, e^t)\, p(e^s, e^t) \\ &= p(r \mid e^s, e^t, T)\, p(T \mid e^s, e^t)\, p(e^t \mid e^s)\, p(e^s) \\ &\approx p(r \mid e^s, e^t)\, p(T \mid e^t)\, p(e^t \mid e^s) \end{aligned} \qquad (1)$$

In equation (1), there are three components: $p(r|e^s,e^t)$ the probability that r is mentioned between e^s and e^t; $p(T|e^t)$ the probability that e^t mentions target type T; $p(e^t|e^s)$ the probability that e^s mentions e^t. This step is very similar to that in the work [3]. The only difference is the estimation for the components.

2.4 Relevant Candidate Tables/Lists Extraction

The information on the web can be organized or represented as several forms such as natural language text, table, list and so on. In the task related entities finding, many target entities are found not only in the natural language text snippet, but also in the tables/lists. So the extraction of relevant tables/lists is very necessary for this task. We define the relevance of the table/list as follows:

- The context of the table/list is relevant to the query
- The table/list should contain several candidate entities extract from the unstructured text.

Here, we simply apply some heuristic methods such as rules to extract the relevant tables/lists. The features used for extraction of tables/lists are very similar to those used in the work. [4].

2.5 Candidate Entities Re-Ranking

2.5.1 Bipartite Graph Construction

We construct a bipartite graph based on the candidate entities and relevant tables/lists. The candidate entities and relevant tables/lists are regarded as the two disjoint sets of graph vertices. If one candidate entity is in a relevant table/list, the vertices corresponding to them will be connected by an edge. For two vertices e_i, l_j of one edge, the weight of the edge is defined as follows：

$$w_{ij} = \begin{cases} 1 \;\; if\ e_i\ is\ connected\ \ to\ l_j \\ 0 \;\; otherwise \end{cases}$$

2.5.2 Reinforcement Learning

Once the bipartite graph is constructed, we apply the reinforcement Learning algorithm over it. Similar to [5], the Iteration equation is as follows：

$$\begin{cases} E_{n+1} = \alpha M_1 L_n + (1-\alpha) E_0 \\ L_{n+1} = \beta M_2 E_{n+1} + (1-\beta) L_0 \end{cases} \tag{2}$$

$$M_1 = A D_L^{-1}$$

$$M_2 = A^T D_E^{-1}$$

E_0 and L_0 are the initial Ranking value vectors of the candidate entities and relevant table/list, respectively. E_n and L_n are the ranking value vectors after n iterations. A is the biadjacency matrix of bipartite graph. A^T is the transpose of A. D_L is the diagonal matrix with its (i;i)-element equal to the sum of the i-th column of A; D_E is the diagonal matrix with its (i;i)-element equal to the sum of the i-th row of A.αand β are the weight. After several iterations, we get the final score for each candidate entity.

2.6 Find homepages for entities

For the homepage finding, we use search engine such as Google to search for top-K urls and some heuristic rules are used to identify the real homepage. Then we search for the corresponding document from the corpus with the url.

3. Experiments

3.1 Our Experiment

In the experiment, our focus was to retrieve the related entities, and not on finding homepages corresponding to the related entities. We computed the retrieval measure over the name of entities.

We evaluated the proposed model on a standard collection from TREC 2010 Entity Track. The collection includes: (1) ClueWeb09 Category A (about 500 million pages) (2) twenty queries of which topic Ids are from 21 to 40 (3) Relevant entities manually found from web for each query using Google.

For each query, the top 100 ranked entities are returned. We adopt the P@10 and nDCG@R as the measure to evaluate the performance. The ranking strategy based on the probabilistic model is chosen as the baseline. The result is as follows:

	mean P@10	mean nDCG@R
Probabilistic Model (Baseline)	0.2325	0.4530
Bipartite Graph Reinforcement Model	0.2975	0.5150

Table 1: The results of our experiment for entity ranking

From the results, we can see that both the average P@10 and average nDCG@R are improved. So the bipartite graph reinforcement model is more effective for the related entity finding than the probabilistic model.

3.1 Official Results

P@10	nDCG_R	MAP	Rprec
0.1277	0.1611	0.0839	0.1305

Table 2 : The official results for our run

The official results are listed in Table 2. Our scores are low. One possible reason is that we fail to find the true homepages though we could find related entities.

4. Conclusion and Future Work

In this paper, we represent a bipartite graph reinforcement model for the relate entity ranking. As is well known, the entities on the web are embedded not only in the natural language text, but also in the table or list. Given a query, both the candidate entities and relevant tables/lists are extracted from web documents. The ranking of candidate entities can benefit from the relevant tables/lists which may contain several target entities. And Vice versa the ranking of relevant tables/lists may be improved if it contains several candidate entities. The mutual reinforcement between the candidate entities and relevant tables/lists will improve the effectiveness of the entity ranking

However, the performance can be further improved with better techniques. In future work, we will investigate other methods for NER and homepage finding. And we will also extract the target entities from the tables/lists with high score to improve the recall.

5. Acknowledgements

We would like to thank Zeying Peng and Xu Chen for their help with preparation for data and development of some useful tools for our task. In addition, our work is supported by Key Program of NSFC 60933005 and NSFC 60903139. 973 Program of China 2007CB311103.

6. References

[1] Krisztian Balog,Arjen P. de Vries,Pavel Serdyukov,Paul Thomas,Thijs Westerveld. Overview of the TREC 2009 Entity Track. In Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009), Gaithersburg, MD, 2009.

[2] C. Fellbaum. WordNet: An Electronic Lexical Database.MIT Press, 1998.

[3] M. Bron, K. Balog, and M. de Rijke. Related Entity Finding Based on Co-Occurance. In Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009), Gaithersburg, MD, 2009.

[4] Y. Wang and J. Hu. A machine learning based approach for table detection on the web. In Eleventh International World Wide Web Conference, 2002.

[5] Xiaoguang Rui, Mingjing Li, Zhiwei Li, Wei-Ying Ma, Nenghai Yu,Bipartite Graph Reinforcement Model for Web Image Annoation, Proceedings of the ACM International Multimedia Conference and Exhibition, 2007